

Lesson 4

Understanding Web Applications

Knowledge Assessment

Multiple Choice

Circle the letter that corresponds to the best answer.

1. You write the following code for your Web page:

```
<html>
  <head>
    <title>Sample Page</title>
    <style type="text/css">
      div
      {
        font-family: Verdana;
        font-size: 9pt;
      }
    </style>
  </head>
  <body>
    <div style=
      "font-weight: bold; font-size: 12pt;">
      Sample Text</div>
  </body>
</html>
```

What would be the style for the text displayed as part of the <div> element?

- a. **Font family: Verdana; font weight: bold; font size: 12pt**
 - b. Font family: Verdana; font weight: bold; font size: 9pt
 - c. Font family: Verdana; font size: 12pt
 - d. Font family: Verdana; font size: 9pt
2. You are developing a mapping Web site that allows users to interactively explore maps using actions such as panning and zooming. You want the Web site to be responsive and accessible in most modern Web browsers. However, you do not want users to have to install additional plug-ins in order to use your Web site. Which of the following technologies should you use to display maps?
- a. HTML
 - b. Server-side programming technology such as ASP.NET
 - c. Adobe Flash
 - d. **JavaScript**
3. Your ASP.NET page contains a page-level variable of Customer type. You want to preserve the value of this variable across page postbacks, but you do not need this variable in any other page in the application. Which of the following state-management techniques is the best way to achieve this?

- a. Query strings
 - b. Cookies
 - c. ViewState**
 - d. Session
4. You are developing a Web application for an online bank. Your application enables users to access their account information and transactions from within a Web browser. When a user logs onto the Web application, you want to show the username and account balance on all pages of the application until the user logs off. You also want this application to be safe from malicious users. Which of the following state-management techniques should you use?
- a. Cookies
 - b. ViewState
 - c. ViewState with encryption
 - d. Session**
5. You are developing a Web form to display weather information. When a user requests the Web form, the form needs to perform some initialization to change its appearance and assign values to some controls. Where should you put the code?
- a. In the PreInit event handler of the Page class
 - b. In the Init event handler of the Page class
 - c. In the Load event handler of the Page class**
 - d. In the PreRender event handler of the Page class
6. You want to display values of C# expressions in an ASP.NET page. Which of the following types of code blocks should you use to enclose the expression?
- a. `<script runat="server">...</script>`
 - b. `<script>...</script>`
 - c. `<%= ... %>`**
 - d. `<form>...</form>`
7. You have developed a timesheet application that will be used by all employees in your company. You used ASP.NET to develop this application and have deployed it on the company's Web server. What must all employees of the company install on their computers before they can access the timesheet application?
- a. .NET Framework Redistributable
 - b. .NET Framework Software Development Kit
 - c. Visual Studio
 - d. A Web browser**
8. Your client application calls a Web service that performs complex, time-consuming calculations. A user complains that while results are being returned, the user interface freezes momentarily. Which approach should you take to solve this issue?
- a. You should install a better processor on the Web server.
 - b. You should install a better processor on the client computer.

- c. You should upgrade to a faster Internet connection.
 - d. You should use asynchronous calls to invoke the Web service.**
9. You have created an ASP.NET Web service that converts one currency into another. One of the methods in your Web service is defined with the following code:

```
public double Convert(double amount,
    string from, string to)
{
    // code to perform currency conversion
}
```

The users of the Web service report that they can set a reference to the Web service but the Convert method is not available to them. What could be the problem?

- a. The .asmx file for the Web service is not available on the Web server.
 - b. The Web service class is not marked with the WebService attribute.
 - c. The Convert method is not marked with the WebMethod attribute.**
 - d. Web services can only expose methods that return text values.
10. You are working on two Visual Studio projects. The first project is a Web service that returns a DataSet object belonging to the System.Data namespace. The second project accesses the Web service created by the first project. Which project in this scenario requires a reference to the System.Data namespace?
- a. The Web service project.
 - b. The client project that accesses the Web service.
 - c. Both the client project and the Web service project.**
 - d. Neither the client project nor the Web service project.

Fill in the Blank

Complete the following sentences by writing the correct word or words in the blanks provided.

- 1. In the HTML anchor tag (<a>), the **href** attribute specifies the target URL.
- 2. You can put CSS code in a separate file and link it to a Web page through use of the HTML **<link>** element.
- 3. The JavaScript code on a Web page is executed on the **client-side**.
- 4. You can use a(n) **<noscript>** element to display a specific message to users when their browser is not running JavaScript.
- 5. You can disable ViewState at the page level by setting the **EnableViewState** attribute of the Page directive to false in the ASP.NET page.
- 6. The **Application** state is used to store data that is used globally throughout an application, as opposed to the **Session** state, which stores data for a user session.
- 7. A Web application is accessed using a(n) **virtual directory** name instead of a physical folder name.
- 8. You must mark classes with the **WebService** attribute to expose them as a Web service.

9. Of all the methods in a Web service class, only those marked with **WebMethod** attributes are exposed as Web service methods.
10. SOAP relies on **XML** as its message format and uses **HTTP** for message transmission.

Competency Assessment

Project 4-1: Using JavaScript and HTML

You are developing a Web page that provides a responsive user interface. You decide to display an image on the page. When the user moves her mouse over the image, the original image is replaced with a new image. Then, when the mouse moves out of the image area, the original image is displayed again. You hope to accomplish this using client-side JavaScript and HTML code. How would you create a Web page that works as described here?

1. **Open Visual Studio. Create a new project based on the ASP.NET Empty Web Application template. Name the project Project04_01.**
2. **Add a new HTML page named default.htm to the project.**
3. **Add two image files (RedBox.png and BlueBox.png) to the project. These image files are a square filled with a solid color. RedBox.png has a square filled with red. Similarly, the BlueBox.png file is a square filled with blue. You can create these files using the Windows Paint application.**
4. **Replace the default code in the HTML page with the following:**

```
<!DOCTYPE HTML PUBLIC
    "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title></title>
    <script type="text/javascript"
      language="javascript">
      <!--
        function SetMouseOverImage() {
          document.boxImage.src = "BlueBox.png";
        }

        function SetMouseOutImage() {
          document.boxImage.src = "RedBox.png";
        }
      -->
    </script>
  </head>
  <body>
```

```

        <p align="center">
            
        </p>
        <p>
            Move the mouse over the red color box above.
            You will see that the image changes to the
            blue color. Now, move the mouse out of the image
            area. The image will change back to its original
            red color.
        </p>
    </body>
</html>

```

5. **Select Debug > Start Debugging (or press F5) to run the project. Move the mouse pointer over the image. You should see the red-color image being replaced by a blue-color image. Then, when you move the mouse pointer out of the image area, the red-color image should be restored.**

Project 4-2: Using Query Strings

You are developing a portion of a Web site that allows users to enter their names and email addresses to subscribe to an email newsletter. Your solution consists of two Web pages. The first page collects the user name and email address and then transfers control to a second page. The second page accepts the name and the email address as query-string parameters and displays a confirmation message to the user. You need to write code for these two pages. What code will you write to accomplish this requirement?

1. **Create a new project based on the ASP.NET Empty Web Application template. Name the project Project04_02.**
2. **Add a new item to the project based on the Web Form template. Name the Web form default.aspx.**
3. **Replace the HTML in the default.aspx file with the following code:**

```

<%@ Page Language="C#" AutoEventWireup="true"
    CodeBehind="default.aspx.cs"
    Inherits="Project04_02._default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
    Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
    transitional.dtd">

```

```
<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
</head>
<body>
    <form id="form1" runat="server">
        <div>
            Enter Name: <asp:textbox ID="NameTextBox"
runat="server" />
        </div>
        <div>
            Enter Email: <asp:textbox ID="EmailTextbox"
runat="server" />
        </div>
        <div />
        <div>
            <asp:Button ID="SubscribeButton" runat="server"
                Text="Subscribe"
                onclick="SubscribeButton_Click"/>
        </div>
    </form>
</body>
</html>
```

- 4. Switch to the code view for the Web page (default.aspx.cs) and replace the code with the following:**

```
using System;

namespace Project04_02
{
    public partial class _default : System.Web.UI.Page
    {
        protected void SubscribeButton_Click(
            object sender, EventArgs e)
        {
            Response.Redirect(
                String.Format(
                    "Confirm.aspx?Name={0}&Email={1}",
                    NameTextBox.Text,
                    EmailTextbox.Text));
        }
    }
}
```

- 5. Add a new item to the project based on the Web Form template. Name the Web form Confirm.aspx.**

6. Switch to the code view for the Web page (Confirm.aspx.cs) and replace the code with the following:

```
using System;

namespace Project04_02
{
    public partial class Confirm : System.Web.UI.Page
    {
        protected void Page_Load(
            object sender, EventArgs e)
        {
            if (Request.QueryString["Email"] != null
                && Request.QueryString["Name"] != null)
            {
                Response.Write(String.Format(
                    "{0}, you are subscribed at: {1}",
                    Request.QueryString["Name"],
                    Request.QueryString["Email"]));
            }
        }
    }
}
```

7. Select Debug > Start Debugging (or press F5) to run the project. Enter a name and email address and click the Subscribe button. Notice that you get transferred to the Confirm.aspx page. Take note of the URL in the Web browser's address bar. The query-string parameters name and email are part of the URL. Also notice that the Confirm.aspx page retrieves that query string to display a customized conformation message.

Proficiency Assessment

Project 4-3: Calling a Web Service Asynchronously

The proxy class generated by Visual Studio for a Web service includes methods for calling the Web service synchronously as well as asynchronously. By default, the application uses the synchronous method. If you prefer asynchronous invocation, you need to call the asynchronous version of the method. The asynchronous versions do not wait for the Web service to return a response and use a callback mechanism to get a response when it is ready. Asynchronous invocation of a Web service may help client applications be more responsive. You want to call the ToLower method of the previously created TextWebService in an asynchronous fashion. What code would you write for asynchronously invoking a Web service?

1. Create a new ASP.NET Web Application project named ToLowerAsync.
2. Add a Web Reference to the TextWebService Web service. Name the reference textWebService.
3. Change the code of the Default.aspx to the following code. Notice here that the Page attribute has an Async property set to true. This property must be set to true before a page can start an asynchronous operation:

```
<%@ Page Title="Home Page" Language="C#"
    AutoEventWireup="true"
    CodeBehind="Default.aspx.cs"
    Async="true"
    Inherits="ToLowerAsync._Default" %>
<html>
    <head><title>TextWebService Client
    </title></head>
    <body>
        <form id="Form1" runat="server">
            <h2>Test Form For TextWebService</h2>
            <p>
                <asp:TextBox ID="TextBox1"
                    runat="server"
                    Text="enter text" />
                <br />
                <asp:Button ID="Button1"
                    runat="server"
                    Text="Async ToLower"
                    onclick="Button1_Click" />
            </p>
            <p>
                <strong>Results from ToLower
                method:</strong><br />
                <asp:Label ID="toLowerLabel"
                    runat="server"
                    Text="Label" ForeColor="Green" />
            </p>
        </form>
    </body>
</html>
```

4. Add the following code to the code-behind for the Web Form:

```
protected void Button1_Click(
    object sender, EventArgs e)
{
    var webService =
```



```
        new textWebService.TextWebService();
        webService.ToLowerCompleted +=
            new textWebService
                .ToLowerCompletedEventHandler(
                    webService_ToLowerCompleted);
        webService.ToLowerAsync(TextBox1.Text);
    }

    void webService_ToLowerCompleted(object sender,
        textWebService.ToLowerCompletedEventArgs e)
    {
        toLowerLabel.Text = e.Result;
    }
}
```

5. **Run the Web application, enter some sample text, and click the Async ToLower button. The final value in the label is displayed by the webService_ToLowerCompleted callback method.**

Project 4-4: Using Session State

You are developing a portion of a Web site that allows users to enter their names and email addresses to subscribe to an email newsletter. Your solution consists of two Web pages. The first page collects the user name and email address, adds them to the session state, and transfers control to a second page. The second page retrieves the name and the email address from the session state and displays a confirmation message to the user. You need to write code for these two pages. What code will you write to accomplish this requirement?

1. **Create a new project based on the ASP.NET Empty Web Application template. Name the project Project04_04.**
2. **Add a new item to the project based on the Web Form template. Name the Web form default.aspx.**
3. **Replace the HTML in the default.aspx file with the following code:**

```
<%@ Page Language="C#" AutoEventWireup="true"
    CodeBehind="default.aspx.cs"
    Inherits="Project04_04._default" %>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0
    Transitional//EN"
    "http://www.w3.org/TR/xhtml1/DTD/xhtml1-
    transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <title></title>
```

```

</head>
<body>
    <form id="form1" runat="server">
        <div>
            Enter Name: <asp:textbox ID="NameTextBox"
            runat="server" />
        </div>
        <div>
            Enter Email: <asp:textbox ID="EmailTextbox"
            runat="server" />
        </div>
        <div />
        <div>
            <asp:Button ID="SubscribeButton" runat="server"
            Text="Subscribe"
            onclick="SubscribeButton_Click"/>
        </div>
    </form>
</body>
</html>

```

4. Switch to the code view for the Web page (default.aspx.cs) and replace the code with the following:

```
using System;
```

```

namespace Project04_04
{
    public partial class _default : System.Web.UI.Page
    {
        protected void SubscribeButton_Click(
            object sender, EventArgs e)
        {
            // Add name and email to the session state
            Session["Name"] = NameTextBox.Text;
            Session["Email"] = EmailTextbox.Text;

            Response.Redirect("Confirm.aspx");
        }
    }
}

```

5. Add a new item to the project based on the Web Form template. Name the Web form Confirm.aspx.
6. Switch to the code view for the Web page (Confirm.aspx.cs) and replace the code with the following:

```
using System;
```

```
namespace Project04_04
{
    public partial class Confirm : System.Web.UI.Page
    {
        protected void Page_Load(
            object sender, EventArgs e)
        {
            if (Session["Email"] != null
                && Session["Name"] != null)
            {
                Response.Write(String.Format(
                    "{0}, you are subscribed at: {1}",
                    Session["Name"],
                    Session["Email"]));
            }
        }
    }
}
```

7. Select Debug > Start Debugging (or press F5) to run the project. Enter a name and email address and click the Subscribe button. Notice that you are transferred to the Confirm.aspx page. Take note of the URL in the Web browser's address bar. The URL does not contain any query-string parameters. Also notice that the Confirm.aspx page retrieves that name and email from the session state to display a customized confirmation message.